AMENDMENTS TO THE CLAIMS

1. (currently amended): A method of operating a flash memory array comprising a plurality of memory blocks, each memory block respectively associated with a counter, the method comprising the steps of:

erasing a selected one of said memory blocks;

responsive to said erasing:

setting the [[a]] counter associated with said selected one of said memory blocks to a first predetermined value; ~~each time a memory block associated with said counter is erased;~~ and

incrementing each [[a]] counter not ~~respectively~~ associated with said selected one of said memory blocks; ~~another memory block in said flash memory array;~~ and

~~refreshing the data in a memory block of said flash memory array~~ when a[[n]] ~~associated~~ counter equals or exceeds a predetermined threshold value,[[.]] always refreshing the data in the memory block associated with said counter by:

storing data from the memory block associated with said counter in another one of said memory blocks.

2. (original): A method as in claim 1, wherein said first predetermined value is zero.

3. (canceled):

4. (currently amended): A method as in claim 1, [[3,]] wherein said refreshing further comprises leaving said data in said different memory block when said refreshing is complete.

5. (currently amended): A method as in claim 1, wherein said refreshing further comprises returning ~~restoring said~~ data from said another one of said memory blocks back to ~~in said~~ the memory block[[.]] associated with the counter.

6. (original): A method as in claim 5, wherein said data is restored in the same locations in a memory block as it is read from.

7. (canceled):

8. (original): A method as in claim 1, wherein refreshing comprises dividing said data into pieces, each of said pieces being refreshed after a distinct, unrelated operation is completed by said flash memory array.

9. (original): A method as in claim 8, wherein each of said pieces is refreshed after a write command is executed by said flash memory array.

10. (original): A method as in claim 1, wherein the values of said counters are stored in a table.

11. (original): A method as in claim 10, wherein said table is stored in said flash memory array.

12. (original): A method as in claim 10, wherein said table is stored in a table in a separate memory storage device.

13. (original): A method as in claim 10, wherein the most significant byte and the second most significant byte of each entry in said table are set to predetermined binary values.

14. (original): A method as in claim 13, wherein said predetermined binary values are 11111111 and 11110110.

15. (canceled):

16. (original): A method as in claim 10, wherein entries in said table contain eight data bytes.

17. (currently amended): A method of operating a flash memory array comprising a plurality of main blocks, each main block comprising a plurality of memory blocks,

each memory block respectively associated with a counter, the method comprising the steps of:

erasing a memory block;

responsive to said erasing:

setting a counter associated with said memory block to a first predetermined value; and

incrementing counters respectively associated with all other memory blocks in the same main block;

for each ~~refreshing all memory blocks that have associated counters~~ counter which equal or exceed a second predetermined value,[[.]] always refreshing a memory block associated with said counter by storing data contained in said memory block in a different memory block of said flash memory array.

18. (currently amended): A method of operating a flash memory array comprising a plurality of main blocks, each main block comprising a plurality of memory blocks, each memory block respectively associated with a counter, the method comprising the steps of:

erasing a memory block;

responsive to said erasing:

setting the counter associated with said memory block to a first predetermined value; and

decrementing the counters associated with all other memory blocks within the same main block;

for each ~~refreshing all memory blocks that have associated counters~~ counter which is less than or equal to ~~or exceed~~ a second predetermined value,[[.]] always refreshing a memory block associated with said counter by storing data from said memory block in a different memory block of said flash memory array.

19. (original): A method as in claim 17 or 18, wherein said first predetermined value is zero.

20. (canceled):

21. (currently amended): A method as in claim 17 or 18, wherein in said step of refreshing, data from blocks being refreshed ~~said data~~ is restored in the same locations it is read from.

22. (currently amended): A method as in claim 17 or 18, wherein said step of refreshing comprises dividing ~~said~~ data in blocks being refreshed into pieces, each of said pieces being refreshed after a distinct, unrelated operation is completed by said flash memory array.

23. (original): A method as in claim 22, wherein each of said pieces is refreshed after a write command is executed by said flash memory array.

24. (currently amended): A method of operating a flash memory array comprising a plurality of main blocks, each main block comprising a plurality of memory blocks, each memory block comprising a plurality of sectors, each sector respectively associated with a counter, the method comprising the steps of:

programming a sector of one of said memory blocks;

responsive to said programming:

setting a counter associated with the programmed sector to a first predetermined value; and ~~each time a sector of a memory block associated with said counter is programmed;~~

incrementing each [[a]] counter ~~respectively~~ associated with each adjacent sector of said one memory block; ~~in said flash memory array;~~ and

always refreshing the data in a sector of the one [[a]] memory block of said flash memory array when an associated counter equals or exceeds a predetermined threshold value[[.]] by storing data contained in that sector to different sector.

25. (original): A method as in claim 24, wherein said first predetermined value is zero.

26. (canceled):

27. (currently amended): A method as in claim 24, [[27,]] wherein said refreshing further comprises leaving said data in said different sector when said refreshing is complete.

28. (currently amended): A method as in claim 24, wherein said refreshing comprises returning said data from said different sector to ~~restoring said data from~~ said sector.

29. (original): A method as in claim 28, wherein said data is restored in the same locations in a sector as it is read from.

30. (canceled):

31. (currently amended): A method as in claim 24, wherein when the counters associated with multiple sectors equal or exceed said predetermined threshold simultaneously, said refreshing comprises refreshing some ~~each~~ sectors before a distinct, unrelated operation is completed by said flash memory array and refreshing some other sectors after said distinct, unrelated operation is completed by said flash memory array.

32. (currently amended): A method as in claim 31, wherein each of said sectors is refreshed after a next write command is executed by said flash memory array.

33. (original): A method as in claim 24, wherein the values of said counters are stored in a table.

34. (original): A method as in claim 33, wherein said table is stored in said flash memory array.

35. (original): A method as in claim 33, wherein said table is stored in a table in a separate memory storage device.

36. (original): A method as in claim 33, wherein the most significant byte and the second most significant byte of each entry in said table are set to predetermined binary values.

37. (original): A method as in claim 36, wherein said predetermined binary values are 11111111 and 11110110.

38. (canceled):

39. (original): A method as in claim 33, wherein entries in said table contain eight data bytes.

40. (currently amended): A method of operating a flash memory array comprising a plurality of main blocks, each main block comprising a plurality of memory blocks, each memory block comprising a plurality of sectors, each sector respectively associated with a counter, the method comprising the steps of:

programming a sector in a memory block;

responsive to said programming:

setting a counter associated with said sector to a first predetermined value; and

incrementing counters respectively associated with all other sectors adjacent to said programmed sector;

always refreshing all sectors that have associated counters which equal or exceed a second predetermined value[[.]] by storing data from a sector in a different sector in said flash memory array.

41. (currently amended): A method of operating a flash memory array comprising a plurality of main blocks, each main block comprising a plurality of memory blocks, each memory block comprising a plurality of sectors, each sector respectively associated with a counter, the method comprising the steps of:

programming a sector in a memory block;

responsive to said programming:

setting a counter associated with said sector to a first predetermined value;

decrementing counters respectively associated with all other sectors adjacent to said programmed sector; and

always refreshing all sectors that have associated counters which are less than or equal to ~~or exceed~~ a second predetermined value[[.]] by storing data from a sector in a different sector in said flash memory array.

42. (original): A method as in claim 40 or 41, wherein said first predetermined value is zero.

43. (canceled):

44. (original): A method as in claim 40 or 41, wherein said data is restored in the same locations it is read from.

45. (currently amended): A method as in claim 40 or 41, wherein when the counter associated with multiple sectors equal or exceed said second predetermined value simultaneously, said refreshing comprises refreshing some ~~each~~ of said sectors before a distinct, unrelated operation is completed by said flash memory array, and refreshing some other sectors after said ~~a~~ distinct, unrelated operation is completed by said flash memory array.

46. (original): A method as in claim 45, wherein each of said pieces is refreshed after a write command is executed by said flash memory array.

47. (currently amended): A flash memory storage device comprising:

a control circuit; and

a data storage area divided into a plurality of memory blocks for storing data and a table for storing a plurality of counters[[;]] values, each one of said counter values being respectively associated with one of said plurality of memory blocks;

wherein when said control circuit erases data stored in one of said memory blocks, said control circuit sets the counter value associated with said erased memory block to a first predetermined value and increments counter[[s]] values respectively associated with other memory blocks; and

wherein when one of said plurality of counter[[s]] values equals or exceeds a predetermined threshold value, said control circuit always refreshes the data in the memory block associated with said counter[[.]] by storing the data stored in said memory block in a different memory block.

48-50. (canceled):

51. (original): A flash memory storage device as in claim 47, wherein said control circuit optimizes said data prior to restoring said data.

52. (currently amended): A flash memory storage device as in claim 47, wherein when a counter value associated with a memory block equals or exceeds said predetermined threshold value, said control circuit divides said data into pieces, said control circuit

refreshing some ~~each~~ of said pieces before performing a distinct, unrelated operation and some other of said pieces after performing said [[a]] distinct, unrelated operation.

53. (currently amended): A flash memory storage device as in claim 52, wherein said control circuit refreshes each of said pieces after performing a next write command.

54. (original): A flash memory storage device as in claim 47, wherein the values of said counters are stored in a table.

55. (original): A flash memory storage device as in claim 54, wherein said table is stored in said flash memory storage device.

56. (original): A flash memory storage device as in claim 54, wherein said table is stored in a separate memory storage device.

57. (original): A flash memory storage device as in claim 54, wherein the most significant byte and the second most significant byte of each entry in said table are set to predetermined binary values.

58. (original): A flash memory storage device as in claim 57, wherein said predetermined binary values are 11111111 and 11110110 respectively.

59. (canceled):

60. (currently amended): A flash memory storage device comprising:

a control circuit; and

a data storage area divided into a plurality of memory blocks, said memory blocks further divided into a plurality of sectors, for storing data and a table for storing a plurality of counter[[s;]] values, each counter value respectively associated with one of said plurality of sectors;

wherein when said control circuit programs one of said sectors, said control circuit sets the counter value associated with said programmed sector to a first predetermined value and increments counter[[s]] values respectively associated with all sectors adjacent to said programmed sector; and

wherein when one of said plurality of counters equals or exceeds a predetermined threshold value, said control circuit always refreshes the data in the sector associated with said counter[[.]] by storing the data stored in said sector in a different sector.

61-63. (canceled):

64. (currently amended): A flash memory storage device as in claim 60, ~~61 or 62,~~ wherein said control circuit optimizes said data prior to restoring said data.

65. (currently amended): A flash memory storage device as in claim 60, wherein when counters associated with multiple sectors equals or exceed said predetermined threshold value simultaneously, said control circuit refreshes some ~~each~~ of said sectors before performing a distinct, unrelated operation and some other of said sectors after performing said [[a]] distinct, unrelated operation.

66. (currently amended): A flash memory storage device as in claim 65, wherein said control circuit refreshes each of said sectors after performing a next write command.

67. (canceled):

68. (currently amended): A flash memory storage device as in claim 60, ~~67,~~ wherein said table is stored in said flash memory storage device.

69. (currently amended): A flash memory storage device as in claim 60, ~~67,~~ wherein said table is stored in a separate memory storage device.

70. (currently amended): A flash memory storage device as in claim 60, ~~67,~~ wherein the most significant byte and the second most significant byte of each entry in said table are set to predetermined binary values.

71. (original): A flash memory storage device as in claim 70, wherein said predetermined binary values are 11111111 and 11110110 respectively.

72. (canceled):

73. (currently amended): A processor circuit comprising:

a processor; and

a flash memory storage device coupled to said processor, said flash memory storage device comprising a plurality of memory storage regions and a plurality of counters, each of said plurality of counters associated with one of said plurality of memory storage regions;

wherein when data stored in one of said memory storage regions is erased, the counter associated with said memory storage region is set to a first predetermined value and the remaining counters are incremented; and

wherein when one of said plurality of counters equals or exceeds a predetermined threshold value, the data in the memory storage region associated with said counter is always refreshed[[.]] by storing data of said memory storage region in a different memory storage region.

74. (original): A processor circuit as in claim 73, wherein said first predetermined value is zero.

75. (canceled):

76. (currently amended): A processor circuit as in claim 73, wherein said flash memory storage device restores said data to the same locations. ~~in said erased memory block.~~

77. (currently amended): A processor circuit as in claim 73, wherein when a memory storage region requires refreshing, said flash memory storage device divides said memory storage region into pieces, said flash memory storage device refreshing some ~~each~~ of said pieces before a distinct, unrelated operation is completed by said flash memory storage device and refreshing some other of said pieces after said [[a]] distinct, unrelated operation is completed by said flash memory storage device.

78. (currently amended): A processor circuit as in claim 77, wherein said flash memory storage device refreshes each of said pieces after new [[a]] write commands are executed by said flash memory storage device.

79. (original): A processor circuit as in claim 73, wherein the values of said counters are stored in a table.

80. (original): A processor circuit as in claim 79, wherein said table is stored in said flash memory storage device.

81. (original): A processor circuit as in claim 79, wherein said table is stored in a separate memory storage device.

82. (canceled):

83. (original): A processor circuit as in claim 79, wherein the most significant byte and the second most significant byte of each table entry are set to a predetermined binary values.

84. (original): A processor circuit as in claim 83, wherein said predetermined binary values are 11111111 and 11110110 respectively.

85. (currently amended): A processor circuit comprising:

a processor; and

a flash memory storage device coupled to said processor, said flash memory storage device comprising a plurality of memory storage regions, said memory storage regions being divided into a plurality of sectors, and a plurality of counters, each of said plurality of counters respectively associated with one of said sectors;

wherein when one of said sectors is programmed, the counter associated with said sector is set to a first predetermined value and all of the sectors adjacent to said programmed sector incremented; and

wherein when one of said plurality of counters equals or exceeds a predetermined threshold value, the data in the sector associated with said counter is always refreshed[[.]] by storing the data stored in said sector to a different sector.

86. (original): A processor circuit as in claim 85, wherein said first predetermined value is zero.

87. (canceled):

88. (currently amended): A processor circuit as in claim 85, wherein said flash memory storage device returns ~~restores~~ said data to ~~the same in~~ said sector.

89. (currently amended): A processor circuit as in claim 85, wherein when the counters associated with multiple sectors equal or exceed said predetermined value simultaneously, said flash memory storage device refreshes some ~~each~~ of said sectors before a distinct, unrelated operation is completed by said flash memory storage device, and some other of said sectors after said [[a]] distinct, unrelated operation is completed by said flash memory storage device.

90. (original): A processor circuit as in claim 89, wherein said flash memory storage device refreshes each of said sectors after a write commands are executed by said flash memory storage device.

91. (original): A processor circuit as in claim 85, wherein the values of said counters are stored in a table.

92. (original): A processor circuit as in claim 91, wherein said table is stored in said flash memory storage device.

93. (original): A processor circuit as in claim 91, wherein said table is stored in a separate memory storage device.

94. (original): A processor circuit as in claim 91, wherein the entries in said table comprise N bytes each, each bit of each entry corresponding to one memory block of N based on said bit's position within said byte.

95. (canceled):

96. (original): A processor circuit as in claim 95, wherein said predetermined binary values are 11111111 and 11110110 respectively.

97. (new): A method of operating a memory array comprising a plurality of memory blocks, each memory block respectively associated with a counter, the method comprising the steps of:

erasing a selected one of said memory blocks;

responsive to said erasing:

setting the counter associated with said selected one of said memory blocks to a first predetermined value; and

incrementing each counter not associated with said selected one of said memory blocks; and

counter equals or exceeds a predetermined threshold value, always refreshing the data in the memory block associated with said counter by:

storing data from the memory block associated with said counter in another one of said memory blocks.

98. (new): A method of operating a memory array comprising a plurality of main blocks, each main block comprising a plurality of memory blocks, each memory block comprising a plurality of sectors, each sector respectively associated with a counter, the method comprising the steps of:

programming a sector of one of said memory blocks;

responsive to said programming:

setting a counter associated with the programmed sector to a first predetermined value; and

incrementing each counter associated with each adjacent sector of said one memory block; and

always refreshing the data in a sector of the one memory block of said memory array when an associated counter equals or exceeds a predetermined threshold value by storing data contained in that sector to different sector.

99. (new): A memory storage device comprising:

a control circuit; and

a data storage area divided into a plurality of memory blocks for storing data and a table for storing a plurality of counters values, each one of said counter values being respectively associated with one of said plurality of memory blocks;

wherein when said control circuit erases data stored in one of said memory blocks, said control circuit sets the counter value associated with said erased memory block to a first predetermined value and increments counter values respectively associated with other memory blocks; and

wherein when one of said plurality of counter values equals or exceeds a predetermined threshold value, said control circuit always refreshes the data in the memory block associated with said counter by storing the data stored in said memory block in a different memory block.

100. (new): A memory storage device comprising:

a control circuit; and

a data storage area divided into a plurality of memory blocks, said memory blocks further divided into a plurality of sectors, for storing data and a table for storing a plurality of counter values, each counter value respectively associated with one of said plurality of sectors;

wherein when said control circuit programs one of said sectors, said control circuit sets the counter value associated with said programmed sector to a first predetermined value and increments counter values respectively associated with all sectors adjacent to said programmed sector; and

wherein when one of said plurality of counters equals or exceeds a predetermined threshold value, said control circuit always refreshes the data in the sector associated with said counter by storing the data stored in said sector in a different sector.